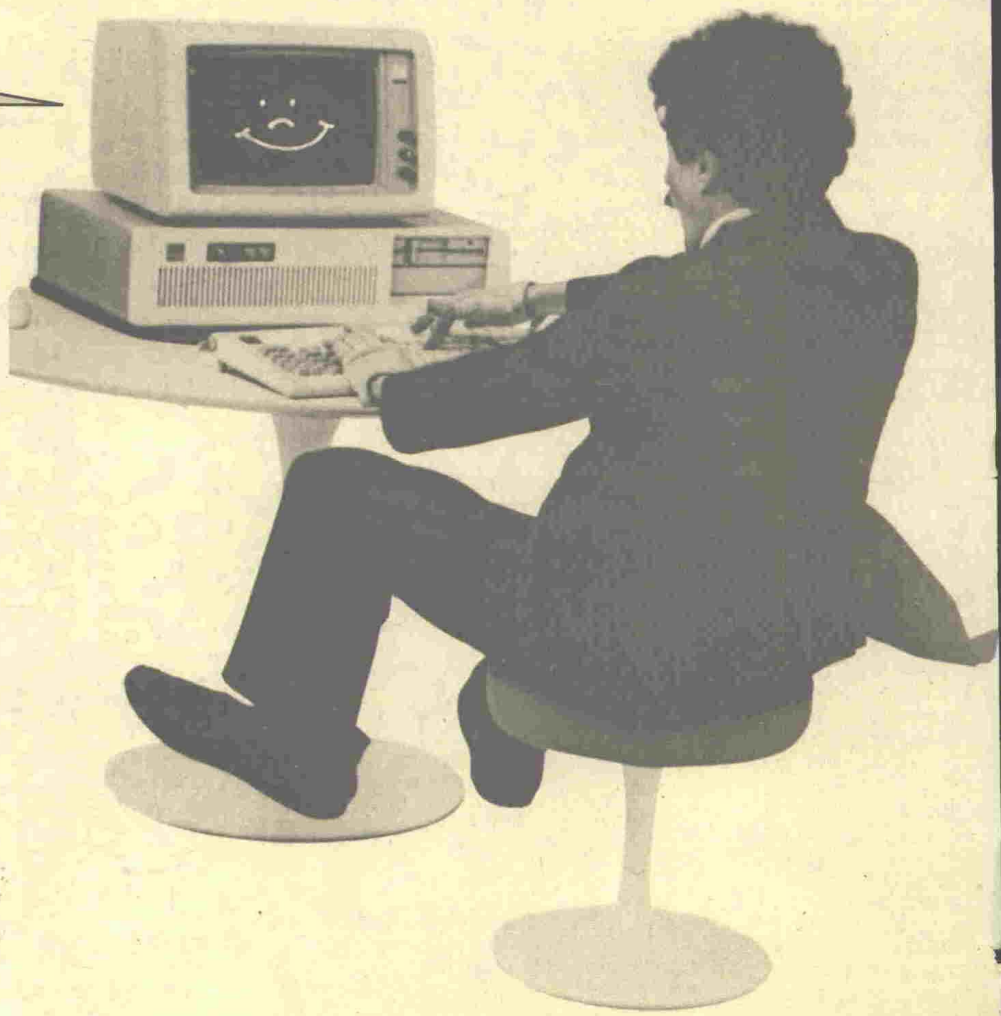
สอน คอมพิวเตอร์ให้พูด

ไมโครโฟนสักตัว...
วงจรอินเตอร์เฟสง่ายๆ กับโปรแกรมสั้นๆ
อีกสองโปรแกรม ผมก็พูดได้แล้วล่ะ...



ชัยแสง เอื้อสกุลเกียรติ

หลาย ๆ ท่านอาจจะทราบแล้วก็ได้นะครับว่าจะทำอย่างไรให้คอมพิวเตอร์พูดได้ และที่จะกล่าวถึงนี้ก็เป็นวิธีหนึ่งซึ่งได้ผลจริง ๆ ซึ่งสามารถทำได้โดยไม่ยากและไม่ต้องใช้อุปกรณ์มากนัก นอกจากโปรแกรมที่สามารถรับข้อมูลเสียงพูดเข้าสู่หน่วยความจำของคอมพิวเตอร์ และโปรแกรมที่สามารถนำข้อมูลเสียงพูดที่อยู่ในหน่วยความจำ มาแปลงกลับให้เป็นเสียงพูดทางลำโพงเท่านั้น

บทความนี้จะใช้คอมพิวเตอร์แอปเปิ้ลทูเป็นหลักในการอ้างอิง ส่วนโปรแกรมการทำงานจะมีทั้งแอปเปิ้ลทูและไอบีเอ็มพีซี สำหรับผู้ที่ใช้เครื่องที่นอกเหนือจากนี้คงต้องเขียนโปรแกรมเองโดยยึดแนวคิดเบื้องต้นเป็นหลัก

แนวคิดเบื้องต้น

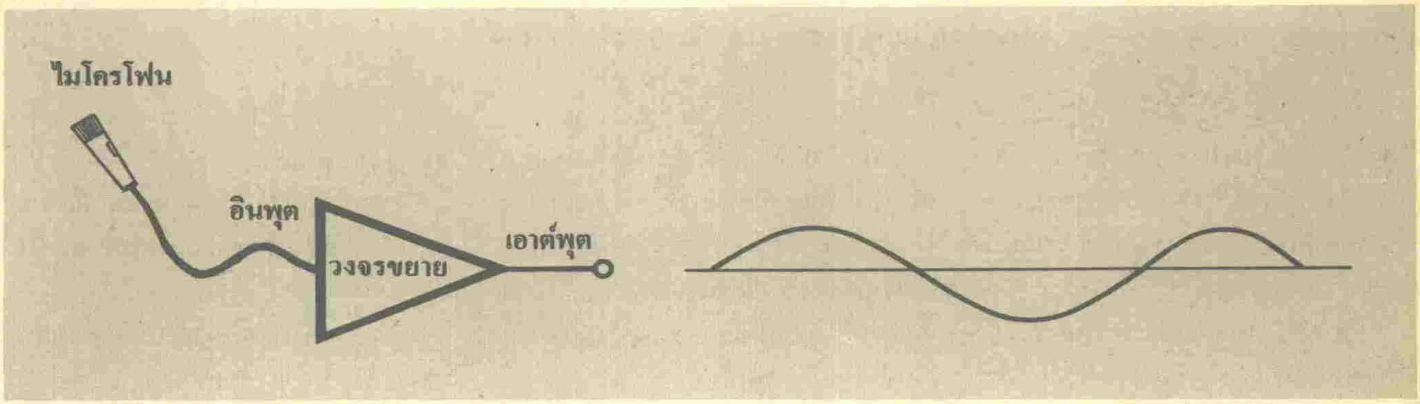
คือ จะทำอย่างไรจึงจะสามารถบันทึกข้อมูลเสียงพูด สู่หน่วยความจำของคอมพิวเตอร์ได้ คำตอบก็คือ ต้องเปลี่ยนเสียงให้เป็นสัญญาณทางไฟฟ้าเสียก่อน ดังรูปที่ 1

จากนั้นจะต้องแปลงสัญญาณเสียงที่ได้มาซึ่งเป็นสัญญาณอะนาลอกเป็นสัญญาณแบบดิจิตอล แล้วจึงบันทึกลงสู่หน่วยความจำต่อไปในรูปของตัวเลขฐานสอง

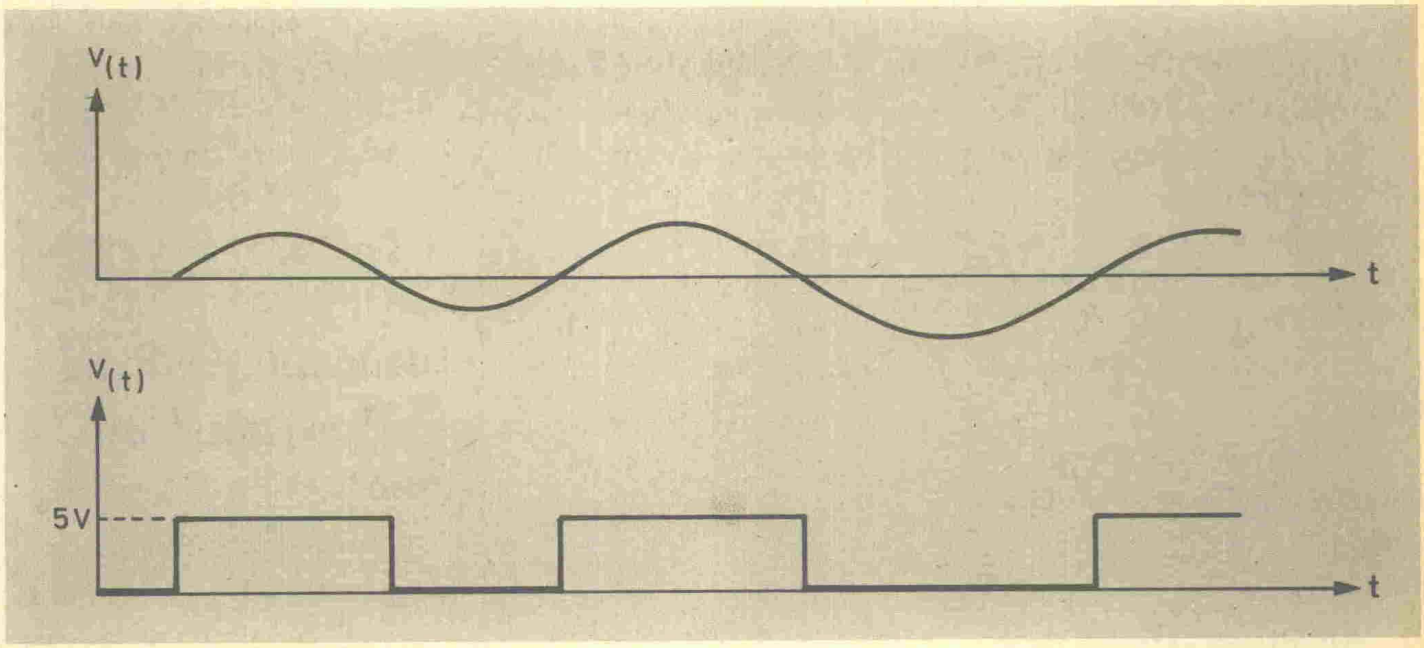
ปัญหาก็คือจะทำให้สัญญาณเสียงพูดนี้เป็นสัญญาณแบบดิจิตอลได้อย่างไร ถ้าดูจากรูปที่ 1 จะพบว่าสัญญาณเสียงพูดนี้เป็นสัญญาณที่มีหลายระดับ และถ้าเราขลิบสัญญาณนี้ให้เป็นสัญญาณที่มีเพียงสองระดับเท่านั้น (อาจจะใช้ DIODEขลิบยอดสัญญาณ) จะได้สัญญาณในรูปที่ 2

ซึ่งถ้าพิจารณาจากรูปสัญญาณทั้งสองนี้ จะพบว่ามีส่วนที่คล้ายคลึงกันอยู่มากและถ้านำสัญญาณนี้ไปขยายแล้วนำออกสู่ลำโพงก็จะเป็นเสียงพูดที่แตกพร่าแต่สามารถฟังรู้เรื่อง เราจะนำสัญญาณที่สองนี้มาบันทึกลงในหน่วยความจำและนำมาเล่นกลับในภายหลังได้ เราจะใช้วิธีสุ่มสัญญาณแล้วนำมาบันทึกลงในหน่วยความจำดังรูปที่ 3

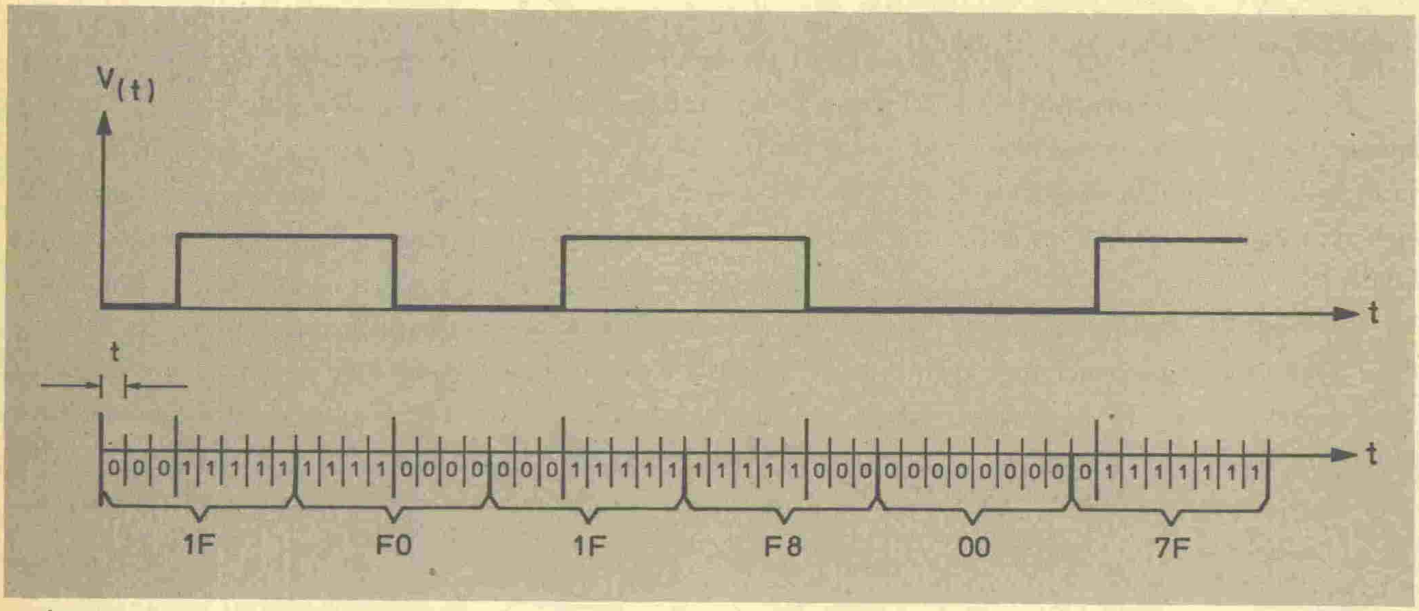
และเมื่อเรารวมข้อมูลที่ได้นี้ 8 บิต เป็น 1 ไบต์ และเก็บในหน่วยความจำก็จะได้สัญญาณเสียงพูดเก็บไว้ในหน่วยความจำตามต้องการ จากรูปถ้าพิจารณาจากอัตราการสุ่มสัญญาณจะพบว่าถ้าคาบเวลา t ของการสุ่มยิ่งน้อยเท่าไร รายละเอียดของเสียงก็จะมากขึ้น จำนวนข้อมูลก็จะมีมากขึ้นและถ้าคาบของการสุ่มสัญญาณมาก รายละเอียดของเสียงที่ได้ก็จะน้อยลงทำให้



รูปที่ 1 การแปลงสัญญาณเสียงเป็นสัญญาณไฟฟ้า



รูปที่ 2 การแปลงสัญญาณอะนาลอกเป็นดิจิตอล



รูปที่ 3 การสุ่มสัญญาณดิจิตอล เพื่อแปลงเป็นข้อมูลฐาน 16 เก็บในหน่วยความจำ

เสียงที่ได้เพี้ยนไปจากความเป็นจริงมากขึ้น จะเห็นได้ว่า สิ่งที่กำหนดความชัดเจนของเสียงพูดก็คือ คาบเวลาหรืออัตราการสุ่มสัญญาณนั่นเอง

สำหรับโปรแกรมที่จะทำให้สัญญาณเสียงพูดนี้ไปปรากฏที่ลำโพงจึงต้องทำตามลำดับขั้นตอนดังนี้

1. นำข้อมูลเสียงพูดมาทีละไบต์
2. ตรวจดูบิตที่มีนัยสำคัญที่สุด (บิต 7)
3. ถ้ามีค่าเป็น 1 ให้ปล่อยไฟไปที่ลำโพง ถ้ามีค่าเป็น 0 ให้ดับไฟที่ลำโพง
4. หน่วงเวลาเท่ากับ t (คาบเวลาของการสุ่มสัญญาณ)
5. เลื่อนบิตไปทางซ้าย 1 บิต
6. ตรวจสอบว่าครบ 8 บิตหรือยัง ถ้ายังไปทำที่ข้อ 2 จนครบ 8 บิต
7. กลับไปทำข้อ 1 ใหม่จนจบข้อมูลเสียงพูด

วิธีที่กล่าวถึงนี้เป็นวิธีที่ไม่สู้จะดีนักเพราะจำนวนข้อมูลเสียงพูดจะมีมากเกินไปเราจึงจะใช้วิธีที่ดีกว่าซึ่งจะกล่าวต่อไปนี้

ถ้าพิจารณาจากข้อมูลเสียงพูดในรูปที่ 3 จะพบว่าถ้าเราใช้คาบเวลา t = 1/4000 วินาที ∴ 1 วินาทีจะได้ข้อมูลเสียงพูด 4000 บิต และถ้าเราให้คาบเวลา t = 1/8000 วินาที ∴ 1 วินาทีก็จะได้ข้อมูลเสียงพูด 8000 บิต ซึ่งเท่ากับ 1000 ไบต์ต่อวินาที ทำให้หน่วยความจำไม่สามารถเก็บข้อมูลเสียงพูดที่ยาว ๆ ได้

ถ้าพิจารณาข้อมูลเสียงพูดที่ได้แล้วแปลงเป็นข้อมูลใหม่ที่มีความสัมพันธ์กับข้อมูลเดิมก็จะได้ข้อมูลใหม่ซึ่งสามารถใช้แทนข้อมูลเสียงพูดเดิมที่มีความยาวลดลงได้ วิธีนั้นก็คือการใช้วิธีนับจำนวนข้อมูลที่เท่ากับ 1 และข้อมูลที่เท่ากับ 0 แล้วจึงบันทึกลงสู่หน่วยความจำต่อไป

จากรูปที่ 3 พบว่า
มีเลข 0 3 ตัว
ตามด้วย 1 9 ตัว
ตามด้วย 0 7 ตัว
ตามด้วย 1 10 ตัว
ตามด้วย 0 12 ตัว
.
.
.
ฯลฯ

ซึ่งสามารถใช้แทนข้อมูลเดิมในรูปที่ 3 ได้ ซึ่งจะทำให้ข้อมูลใหม่ที่ได้มานี้สั้นลงกว่าเดิมมาก

สำหรับโปรแกรมที่จะทำให้เสียงพูดนี้ปรากฏที่ลำโพงต้องทำตามลำดับขั้นตอนดังนี้

1. นำข้อมูลเสียงพูดที่เป็นจำนวนของ "1" มา 1 ช่วง
2. หน่วงเวลาเท่ากับข้อมูลเสียงพูด × คาบเวลาในการสุ่มสัญญาณ
3. ปล่อยไฟไปที่ลำโพง
4. นำข้อมูลเสียงพูดที่เป็นจำนวนของ "0" มาอีก 1 ช่วง
5. หน่วงเวลาเท่ากับข้อมูลเสียงพูด × คาบเวลาในการสุ่มสัญญาณ
6. ดับไฟที่ลำโพง
7. กลับไปทำข้อ 1 ใหม่ จนจบข้อมูลเสียงพูด

โปรแกรมและอุปกรณ์ที่ใช้ในการเก็บข้อมูลเสียงพูด

สำหรับแอปเปิ้ลทำได้ดังนี้คือ

เนื่องจากแอปเปิ้ลมีพอร์ตที่สามารถใช้ในงานนี้ได้อยู่แล้วคือ Cassette data in ดังรูปที่ 4

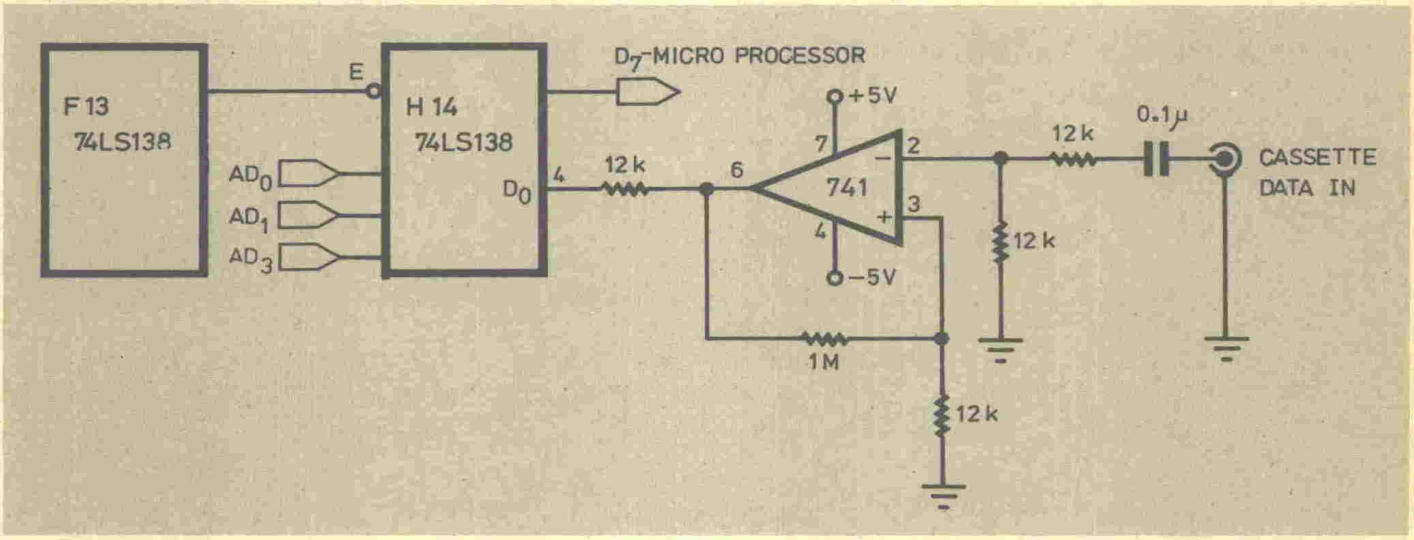
พิจารณาจากวงจรอินเตอร์เฟสของแอปเปิ้ลพบว่าเมื่อให้สัญญาณเสียงเข้าที่ Cassette data in เราสามารถเก็บข้อมูลของสัญญาณเสียงนี้ได้ที่แอดเดรส $C060 โดยที่มีออปแอมป์ เบอร์ 741 ทำหน้าที่เป็น Comparator เมื่อสัญญาณเสียงพูดเข้ามาที่ขา 2 (อินพุต) ของออปแอมป์เบอร์ 741 ก็จะปรากฏเป็นสัญญาณที่ขลิบเป็นรูปสี่เหลี่ยมที่ขา 6 (เอาต์พุต) ของออปแอมป์และก็จะถูกนำเข้าสู่ซีพียู ที่ดาต้าบัสบิต 7 เราก็เพียงแต่เขียนโปรแกรมที่ทำตามลำดับขั้นตอนที่กล่าวมาแล้วข้างต้นดังรูปที่ 5 สำหรับไอบีเอ็มพีซี หรืออื่น ๆ ก็สามารถทำได้เช่นเดียวกัน ในที่นี้จะกล่าวถึงเฉพาะไอบีเอ็มเท่านั้นดังต่อไปนี้คือ

1. ก่อนอื่นจะต้องสร้างวงจรอินเตอร์เฟสดังรูปที่ 6

นำ OUTPUT ไปต่อเข้าที่ GAME CONTROLCARD ที่ D-SHELL CONNECTOR ขา 2, 7, 10, หรือ 14 ขาใดขาหนึ่งดังรูปที่ 7

2. เขียนโปรแกรมที่สามารถทำตามลำดับขั้นตอนที่กล่าวมาแล้วดังรูปที่ 8

รูปที่ 4 คาสเซตต์อินเตอร์เฟสพอร์ตของแอปเปิ้ล



รูปที่ 5 โปรแกรมแอสเซมบลีสำหรับแอปเปิ้ล

(ก) โปรแกรมควบคุมการบันทึกข้อมูลเสียงพูด

```
SOURCE FILE: VOICE DATA CUPTURE
----- NEXT OBJECT FILE NAME IS VOICE DATA CUPTURE.OBJO
5000:              1          ORG  $5000
5000:              2 ********************************
5000:              3 *          *
5000:              4 * VOICE DATA       *
5000:              5 * CUPTURE ROUTINE      *
5000:              6 *          *
5000:              7 ********************************
5000:              8 *          *
C060:              9 TAPEIN  EQU  $C060
0050:             10 COUNT   EQU  $50
0051:             11 VODATAL EQU  $51
0052:             12 VODATAH EQU  $52
5000:             13 *          *
5000:             14 ********************************
```

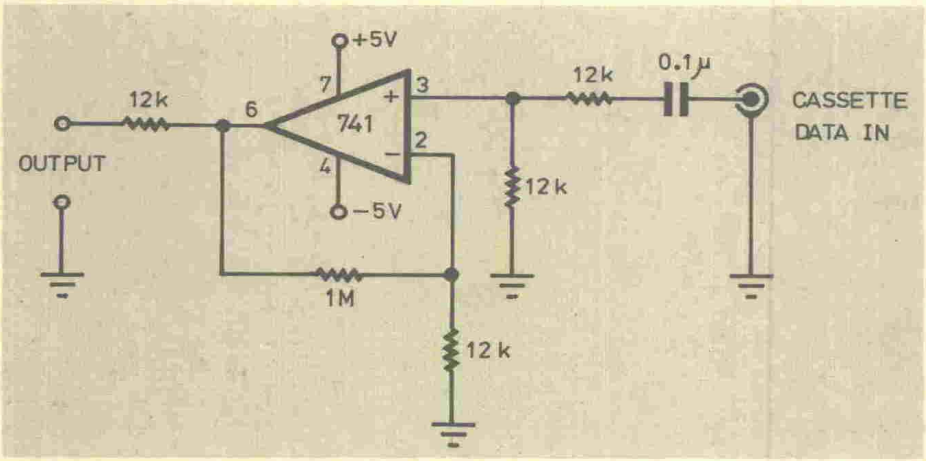
```
5000:              15 *          *
5000:4C 34 50      16            JMP  START
5003:A2 0A         17 DELAY      LDX  #$0A       ;SUBROUTINE DELAY
5005:CA            18 DELAY1     DEX               ;FOR SAMPLING
5006:D0 FD         19            BNE  DELAY1     ;TIME
5008:60            20            RTS
5009:              21 *          *
5009:              22 ********************************
5009:              23 *          *
5009:AD 60 C0      24 READ       LDA  TAPEIN     ;READ VOIVE DATA
500C:10 04         25            BPL  CON1       ;IF HIGHT
500E:20 16 50      26            JSR  COUNTO     ;IF LOW
5011:60            27            RTS
5012:20 25 50      28 CON1       JSR  COUNT1
5015:60            29            RTS
5016:              30 *          *
5016:              31 ********************************
5016:              32 *          *
5016:A9 00         33 COUNTO     LDA  #$00
5018:85 50         34            STA  COUNT      ;CLEAR COUNT
501A:E6 50         35 CCOUNTO    INC  COUNT      ;START COUNT
501C:20 03 50      36            JSR  DELAY
501F:AD 60 C0      37            LDA  TAPEIN
5022:30 F6         38            BMI  CCOUNTO
5024:60            39            RTS
5025:              40 *          *
5025:              41 ********************************
5025:              42 *          *
5025:A9 00         43 COUNT1     LDA  #$00
5027:85 50         44            STA  COUNT      ;CLEAR COUNT
5029:E6 50         45 CCOUNT1    INC  COUNT      ;START COUNT
502B:20 03 50      46            JSR  DELAY
502E:AD 60 C0      47            LDA  TAPEIN
5031:10 F6         48            BPL  CCOUNT1
5033:60            49            RTS
5034:              50 *          *
5034:              51 ********************************
5034:              52 *          *
5034:A9 00         53 START      LDA  #$00
5036:85 51         54            STA  VODATAL
5038:A9 60         55            LDA  #$60       ;START OF VOICE DATA
503A:85 52         56            STA  VODATAH
503C:A0 00         57 CUPTURE    LDY  #$00
503E:20 09 50      58 LOOPCUP    JSR  READ
5041:A5 50         59            LDA  COUNT
5043:91 51         60            STA  (VODATAL),Y
5045:C8            61            INY
5046:D0 F6         62            BNE  LOOPCUP
5048:E6 52         63            INC  VODATAH
504A:A5 52         64            LDA  VODATAH
504C:C9 75         65            CMP  #$75
504E:D0 EC         66            BNE  CUPTURE
5050:60            67            RTS
5051:              68 *          *
5051:              69 ********************************

*** SUCCESSFUL ASSEMBLY: NO ERRORS
```

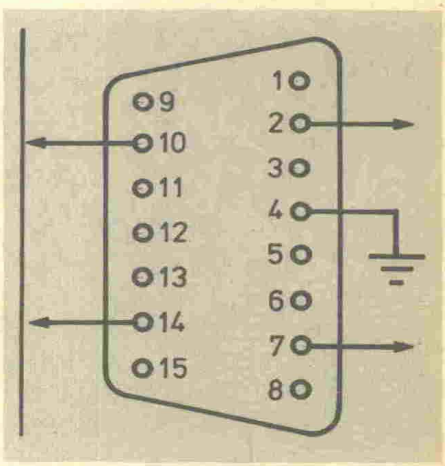
(ข) โปรแกรมควบคุมการเล่นกลับ

```
SOURCE FILE: VOICE DATA RETRIVEL
----- NEXT OBJECT FILE NAME IS VOICE DATA RETRIVEL.OBJO
0300:              1          ORG  $300
0300:              2 ********************************
0300:              3 *          *
0300:              4 * VOICE DATA        *
0300:              5 * RETRIVEL ROUTINE      *
0300:              6 *          *
0300:              7 ********************************
0300:              8 *          *
C030:              9 SPKR     EQU  $C030
6000:             10 VOICE    EQU  $6000
0300:             11 *          *
0300:             12 ********************************
0300:             13 *          *
0300:AE 00 60     14 START    LDX  VOICE
0303:F0 1A        15          BEQ  EXIT
0305:A0 0A        16 DELAY    LDY  #$0A
0307:88           17 DELAY1   DEY
0308:D0 FD        18          BNE  DELAY1
030A:CA           19          DEX
030B:D0 F8        20          BNE  DELAY
030D:AD 30 C0     21          LDA  SPKR
0310:EE 01 03     22          INC  $301
0313:D0 EB        23          BNE  START
0315:EE 02 03     24          INC  $302
0318:AD 02 03     25          LDA  $302
031B:C9 C0        26          CMP  #$C0
031D:D0 E1        27          BNE  START
031F:60           28 EXIT     RTS
0320:             29 *          *
0320:             30 ********************************

*** SUCCESSFUL ASSEMBLY: NO ERRORS
```



รูปที่ 6 วงจรอินเตอร์เฟสที่ต่อเพิ่มเติมสำหรับไอบีเอ็ม



รูปที่ 7 จุดต่อบนคอนเน็คเตอร์ของ GAME CONTROLLER

รูปที่ 8 โปรแกรมแอสเซมบลีสำหรับไอบีเอ็ม

(ก) โปรแกรมควบคุมการบันทึกเสียง

```
TITLE VOICE DATA CUPTURE ROUTINE
;
DSEG    SEGMENT  PARA 'DATA'
VOICE   DB       4096 DUP(0)
DSEG    ENDS
;
CSEG    SEGMENT  PARA 'CODE'
        ASSUME  CS:CSEG,DS:DSEG
        MOV     AX,CSEG     ;SETUP CODE SEGMENT REG
        MOV     CS,AX
        MOV     AX,DSEG     ;SETUP DATA SEGMENT REG
        MOV     DS,AX
START   PROC    NEAR
        MOV     BX,0        ;RESET INDEX = 0
READ:   IN      AL,201H     ;READ VOICE DATA
        AND     AL,00010000B ;CHECK HIGHT OR LOW
        JNZ     TREAD1
        CALL    READ0         ;COUNT LOW
        INC     BX            ;INCREASE INDEX
        CMP     BX,4095       ;THIS IS END OF DATA
        JZ      END
        JMP     READ          ;CONTINUE
TREAD1: CALL    READ1         ;COUNT HIGHT
        INC     BX            ;INCREASE INDEX
        CMP     BX,4095
        JZ      END
        JMP     READ
END:    INT     20H           ;EXIT
START   ENDP
;
READ0   PROC    NEAR
RD0:    INC     VOICE[BX]     ;COUNT LOW
        CALL    DELAY         ;SAMPLING TIME DELAY
        IN      AL,201H       ;THEN READ
        AND     AL,00010000B ;CHECK HIGHT OR LOW
        JZ      RD0
        RET
READ0   ENDP
;
READ1   PROC    NEAR
RD1:    INC     VOICE[BX]     ;COUNT HIGHT
        CALL    DELAY         ;SAMPLING TIME DELAY
        IN      AL,201H       ;THEN READ
        AND     AL,00010000B ;CHECK HIGHT OR LOW
        JNZ     RD1
        RET
READ1   ENDP
;
DELAY   PROC    NEAR
        MOV     CL,52
DELAY1: DEC     CL
        JNZ     DELAY1
        RET
DELAY   ENDP
CSEG    ENDS
```

(ข) โปรแกรมควบคุมการเล่นกลับ

```
TITLE VOICE DATA RETRIVEL ROUTINE
;
DSEG    SEGMENT  PARA 'DATA'
VOICE   DB       4096 DUP(?)
DSEG    ENDS
;
CSEG    SEGMENT  PARA 'CODE'
        ASSUME  CS:CSEG,DS:DSEG
        MOV     AX,CSEG      ;SETUP CODE SEGMENT REG
        MOV     CS,AX
        MOV     AX,DSEG      ;SETUP DATA SEGMENT REG
        MOV     DS,AX
START   PROC    NEAR
        MOV     BX,OFFSET VOICE
BEGIN:  MOV     AL,[BX]      ;GET VOICE DATA
        CALL    DDELAY       ;DELAY
        MOV     AL,02H       ;OPEN SPEAKER
        OUT     61H,AL
        INC     BX           ;NEXT VOICE DATA
        CMP     BX,4095      ;IS END OF DATA
        JE      END          ;YES EXIT
        MOV     AL,[BX]      ;NO CONTINUE
        CALL    DDELAY       ;DELAY
        MOV     AL,00H       ;CLOSE SPEAKER
        OUT     61H,AL
        INC     BX           ;NEXT VOICE DATA
        CMP     BX,4096
        JNE     BEGIN        ;NO CONTINUE
END:    INT     20H
START   ENDP
;
DDLAY   PROC    NEAR
XXX:    CALL    DELAY        ;DELAY FOR
        DEC     AL           ;VOICE DATA *SAMPLING TIME
        JNZ     XXX
        RET
DDLAY   ENDP
;
DELAY   PROC    NEAR
        MOV     CL,52
DELAY1: DEC     CL
        JNZ     DELAY1
        RET
DELAY   ENDP
CSEG    ENDS
```